REMARKS

The Office Action of December 2, 2004 has been carefully considered. In response thereto, the specification and claims have been amended as set forth above. Reconsideration and allowance in view of the foregoing amendments and the following remarks is respectfully requested.

Replacement drawing sheets are submitted herewith as required.

As to the Declaration, Applicant is not aware of any requirement that the Inventor's Signature must correspond letter-for-letter with the full name of the inventor as listed.

Applicant extends thanks to Examiner Rizzuto for a thorough examination.

Claims 1, 6, 8 and 10 were rejected as being anticipated by Mahin, and claims 2, 3 and 9 were rejected as being unpatentable over the Mahin in view of Mohamed. Claim 4 was rejected as being unpatentable over Mahin in view of Miller. Claims 5 and 11 were rejected as being unpatentable over Mahin in view of Keller. Claim 7 was rejected as being unpatentable over Mahin in view of Sakhin. Claim 12 was rejected as being unpatentable over Mahin in view of Keller and further in view of Miller. The claims have been amended to more clearly distinguish over the cited references. Reconsideration is respectfully requested.

In particular, the claims have been amended to recite that information that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, is inserted *at compile time*. No such feature is taught or suggested by the cited references.

Mahin addresses the problem of dealing with the fetching of variable length instructions - composed of a variable number of bytes - in the cache of a superscalar processor. The issue with variable length instructions in this setup is that an instruction could be located across two cache lines, which would normally lead to a penalty in fetching such an instruction from the cache, since it would involve fetching two cache lines before the complete instruction is available. Because the instructions are variable length one cannot predict where the next instruction starts since this can only be computed when the instructions are decoded. To solve these issues, a mechanism is proposed in which whenever an instruction is fetched from the cache for the first time, the bytes in the instruction are tagged with the information obtained during decoding, that is, information on where an instruction starts in a cache line, for instance. Using this information, the second time the same instruction is fetched one can predict which lines to (pre)fetch and where to start decoding, and no penalty will be incurred. The proposed solution particularly stresses its applicability in the presence of self-modifying code in an out-of-order (superscalar) processor.

Hence in Mahin the marking is a kind of tagging mechanism that occurs at RUN TIME inside the cache, the tagging is NOT part of the instruction format. *See* Mahin column 4, lines 5-16. Also, see column 6, lines 53-63. As a result, the first time an instruction is fetched from the cache, the tagging is not present and can therefore be of no advantage.

In the present invention, by contrast, information that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, is inserted *at compile time.* As a result, the information is available and can be taken advantage of the first the an instruction is fetched. Furthermore, the complexity of run-time operations is reduced.

Accordingly, claims 1 and 8 are believed to patentably define over the cited references.

Dependent claims 2-7 and 9 are also believed to add novel and patentable subject matter to their respective independent claims. Withdrawal of the rejection and allowance of claims 1-9 is respectfully requested.

Respectfully submitted,

Michael J. Ure, Reg. 33,089

Dated: March 9, 2005